

(19)日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11)特許出願公表番号

特表2002-533787

(P2002-533787A)

(43)公表日 平成14年10月8日(2002.10.8)

(51)Int.Cl.  
G 0 9 C 1/00

識別記号  
6 2 0

F I  
G 0 9 C 1/00

テーコード(参考)  
6 2 0 A 5 J 1 0 4  
6 2 0 Z

審査請求 未請求 予備審査請求 有 (全 50 頁)

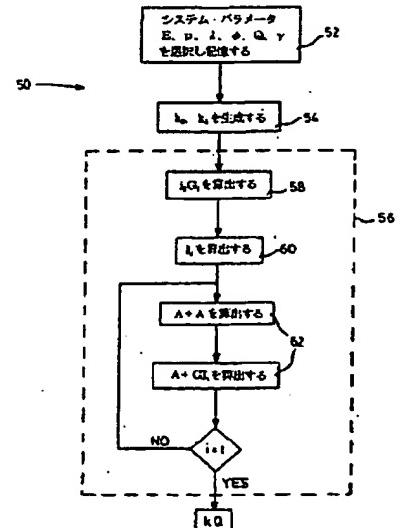
(21)出願番号 特願2000-591498(P2000-591498)  
(86) (22)出願日 平成11年12月23日(1999.12.23)  
(85)翻訳文提出日 平成13年6月22日(2001.6.22)  
(86)国際出願番号 PCT/CA99/01222  
(87)国際公開番号 WO00/39668  
(87)国際公開日 平成12年7月6日(2000.7.6)  
(31)優先権主張番号 2,257,008  
(32)優先日 平成10年12月24日(1998.12.24)  
(33)優先権主張国 カナダ(CA)

(71)出願人 サーティコム コーポレーション  
カナダ、オンタリオ州 エル4ダブリュー  
5エル1、ミシソーガ フォース フロ  
ア、エクスプローラー ドライブ 5520  
(72)発明者 ガラント、ロバート  
カナダ国 エル5エム 5エヌ1 オンタ  
リオ州、ミシソーガ、ローズブッシュ ロ  
ード 4788  
(72)発明者 ランパート、ロバート、ジェイ  
カナダ国 エヌ3シー 3エヌ3 オンタ  
リオ州、ケンブリッジ、ホルム ストリー  
ト 63  
(74)代理人 弁理士 平木 祐輔 (外2名)  
最終頁に続く

(54)【発明の名称】 楕円曲線上での暗号操作の速度を高める方法

(57)【要約】

本発明は、楕円曲線点 $Q(x, y)$ とスカラー $k$ との乗算の速度を高める方法であつて、 $q$ が、楕円曲線上のすべての点 $Q(x, y)$ について $\varphi(Q) = \lambda \cdot Q$ である自己準同型写像 $\varphi$ が存在するような素数べき乗である、有限体 $\mathbb{F}_q$ の上の楕円曲線を選択するステップと、スカラー $k$ のより小さな表現 $l_k$ と写像 $\varphi$ の組合せを使用して楕円曲線点 $Q$ のスカラー倍数を算出するステップとを含む方法を提供する。



【特許請求の範囲】

【請求項 1】 楕円曲線点 $Q(x, y)$ にスカラーを乗じて点 $kQ$ を提供する方法であつて、

a)  $\lambda$ を整数として、楕円曲線上のすべての点 $Q(x, y)$ について $\psi(Q) = \lambda \cdot Q$ である自己準同形写像 $\psi$ が存在するよう、有限体 $F$ の上の楕円曲線を選択するステップと、

b) 前記スカラー $k$ の表現を成分 $k_j$ と前記整数 $\lambda$ の組合せとして確立するステップと、

c) 前記表現と前記点 $Q$ を組み合わせて $kQ$ に対応する倍数の複合表現を形成するステップと、

d)  $kQ$ の前記複合表現から前記点 $kQ$ に対応する値を算出するステップとを含むことを特徴とする方法。

【請求項 2】 前記各成分 $k_j$ が、前記スカラー $k$ よりも短いことを特徴とする、請求項 1 に記載の方法。

【請求項 3】 前記成分 $k_j$ がまず選択され、その後組み合わされて前記スカラー $k$ が形成されることを特徴とする、請求項 1 に記載の方法。

【請求項 4】 前記表現が

【数 1】

$$k_i = \sum_{i=0}^{n-1} k_i x^i \bmod n$$

の形式であり、 $n$ が楕円曲線上の点の数であることを特徴とする、請求項 1 に記載の方法。

【請求項 5】 前記表現が $k_0 + k_1$ の形式であることを特徴とする、請求項 4 に記載の方法。

【請求項 6】 前記スカラー $k$ が所定の値および前記成分 $k$ を有することを特徴とする、請求項 1 に記載の方法。

【請求項 7】 前記倍数 $kQ$ の前記値が、同時多重加算を使用して算出されることを特徴とする、請求項 3 に記載の方法。

【請求項 8】 前記同時多重加算において使用される一群の項 $G_1$ が事前に算

出されることを特徴とする、請求項7に記載の方法。

【請求項9】 体Fの短基底ベクトル( $u_0, u_1$ )を得て、ベクトルvを(k, 0)として指定し、vを正規直交基底から( $u_0, u_1$ )基底に変換し、ベクトルvを表す分数 $f_0, f_1$ を得て、前記分数をkに適用しベクトルzを得て、ベクトルvの効率的な等価物v'を算出し、ベクトルv'の成分をkQの複合表現で使用することによって、前記成分 $k_j$ が得られることを特徴とする、請求項6に記載の方法。

【請求項10】 Qを曲線上の点として、整数kを有し、秘密鍵および公開鍵kQを提供する鍵対を楕円曲線暗号システムにおいて生成する方法であって、

a) 楕円曲線上のすべての点Q(x, y)について $\psi(Q) = \lambda \cdot Q$ であり、 $\lambda$ が整数である自己準同形写像 $\psi$ が存在するように、有限体Fの上の楕円曲線を選択するステップと、

b) 前記スカラーkの表現を成分 $k_j$ と前記整数 $\lambda$ の組合わせとして確立するステップと、

c) 前記表現と前記点Qを組み合わせてkQに対応する倍数の複合表現を形成するステップと、

d) kQの前記複合表現から前記点kQに対応する値を算出するステップとを含むことを特徴とする方法。

【請求項11】 請求項2から9のいずれか一項に記載の方法を含むことを特徴とする、請求項10に記載の方法。

**【発明の詳細な説明】****【0001】****【発明の属する技術分野】**

本発明は、橍円曲線を利用して暗号システムで計算を実行する方法に関する。

**【0002】****【従来の技術】**

公開鍵データ通信システムを使用して一対の通信者との間で情報を転送することができます。交換される情報の少なくとも一部は、送信側により所定の演算によって暗号化され、受信側は、送信側の演算に対して相補的な演算を実行してこの情報を復号することができます。

**【0003】**

各通信者は、秘密鍵と、秘密鍵に数学的に関係づけられた公開鍵とを有する。この関係は、公開鍵の知識から秘密鍵を特定することができないような関係である。各鍵は、転送すべきデータを暗号化するために、あるいはデータが真正であることを検証できるように署名を添付するために、データの転送時に使用される。

**【0004】**

暗号化の場合、一方の通信者は、受信側の公開鍵を使用してメッセージを暗号化し、受信側に送信する。次いで、受信側は秘密鍵を使用してメッセージを復号する。

**【0005】**

一方の当事者の公開鍵を他方の当事者の秘密鍵と組み合わせることによって共通鍵を生成することもできる。このような場合には通常、各当事者の長期鍵が破られるのを回避するため、通信セッションごとに新しい秘密鍵およびこれに対応する公開鍵が生成され、これらの鍵は通常、セッション鍵または短命鍵と呼ばれる。

**【0006】**

したがって、メッセージの交換および公開鍵の生成では、暗号化システム $Z^*_p$ において整数mod p ( $p$ は素数) の有限体を利用する際に指數演算を伴い、あるいは

はシステムが楕円曲線を利用する際に同様な点乗算演算を伴う相当量の計算が行われる。楕円曲線システムでは、秘密整数  $k$  を生成し、シード点  $Q$  で点乗算を実行して短命公開鍵  $kQ$  を形成することによって、短命鍵対が得られる。同様に、共通短命セッション鍵を生成する場合は、公開鍵  $k_3Q$ 、すなわち、曲線上の点に他方の通信者の秘密整数  $k_b$  を乗じる必要があり、したがって、この場合も点乗算が必要になる。

#### 【0007】

メッセージに署名する場合にも、送信側が自分の秘密鍵をメッセージに適用することを除いて、同様な手順が使用される。この場合、任意の受信側が、送信側の公開鍵を使用してメッセージを復元し検証することができる。

#### 【0008】

このような方式を実施するための様々なプロトコルが存在しており、そのうちのいくつかは広く使用されている。しかし、それぞれの場合に、送信側は転送すべき情報に署名するために計算を実行する必要があり、受信側は、署名された情報を検証するために計算を実行する必要がある。

#### 【0009】

典型的な実行形式において、署名成分は以下の形式を有する。

$$s = ae + k \pmod{n}$$

上式で、楕円曲線暗号化システムにおいて、

$P$  は、基本曲線上の、システムの定義済みパラメータである点であり、

$k$  は、短期秘密鍵またはセッション鍵として選択されたランダム整数であり、

$R = kP$  は、これに対応する短期公開鍵であり、

$a$  は、送信側の長期秘密鍵であり、

$Q = aP$  は、これに対応する送信側の公開鍵であり、

$e$  は、メッセージ  $m$  および短期公開鍵  $R$  の、SHA-1ハッシュ関数などの安全ハッシュであり、および

$n$  は曲線の次数である。

#### 【0010】

送信側は、 $m$ 、 $s$ 、および  $R$  を含むメッセージを受信側に送信し、署名は、 $R$

に対応すべき値 $R^1 = (sP - eQ)$ を算出することによって検証される。算出された値が対応する場合、署名は検証されたことになる。

#### 【発明が解決しようとする課題】

##### 【0011】

検証を実行する場合、点乗算を計算し、それぞれ計算が複雑な $sP$ および $eQ$ を得る必要がある。この場合、受信側が適切な計算力を有する場合には、特に問題はないが、セキュアトークン・アプリケーションや「スマート・カード」アプリケーションのように受信側の計算力が限られている場合、このような計算によって検証プロセスに遅延が生じる。

##### 【0012】

したがって、キー生成プロトコルおよび署名プロトコルは、大量の計算を必要とする場合がある。暗号化が普及するにつれて、より高速であり、スマート・カードや無線装置に見られるように限られた計算力を活用する暗号化システムを実現する要求が高まっている。

##### 【0013】

楕円曲線暗号化（ECC）は、この計算問題を解決する。ECCでは、鍵および証明書のサイズを縮小することができ、それによって、必要なメモリが低減し、コストが著しく節約される。ECCは、コストを著しく削減することができるだけでなく、次世代アプリケーションにおけるスマート・カードの普及を促進する。また、ECCアルゴリズムによって鍵サイズを縮小することができるが、より大きな鍵を用いる他のアルゴリズムと同じレベルのセキュリティが維持される。

##### 【0014】

しかし、暗号化装置の低い生産コストを維持しながら情報転送速度を高めるために、依然として、鍵に対する計算をより高速に行う必要がある。

##### 【0015】

楕円曲線上の点の倍数を算出することは、楕円曲線暗号化で最も頻繁に実行される計算の1つである。このような計算の速度を高める1つの方法は、事前計算された点の倍数のテーブルを使用することである。この技法は、点が事前にわかっているときにさらに有用である。しかし、未知の点の倍数が必要になる場合が

ある（たとえば、ECDSA検証）。したがって、点乗算を容易にするシステムおよび方法が必要である。

#### 【0016】

##### 【課題を解決するための手段】

一般に、本発明は、スカラー  $k$  を、成分  $k_i$  と、基本曲線で自己準同形写像から導かれる整数  $\lambda$  との組合せとして表す。

#### 【0017】

この方法は、有限体の上に写像される複素乗算を有する楕円曲線 (EC) が与えられた場合、複素乗算写像と点  $Q$  に  $\lambda$  を乗じることが等しくなる二次方程式の解  $\lambda$  が存在する、という考えに基づく方法である。 $\lambda$  を整数とみなして EC 乗算を実行するのと比べて、複素乗算写像を介して  $\lambda Q$  を算出する方がコストが低いことが多い。実際には、他のスカラー ( $\lambda$  以外) による点乗算が必要になる。乗算写像を使用して点の他の倍数を算出できることも示す。

#### 【0018】

本発明によれば、楕円曲線点  $Q(x, y)$  にスカラー  $k$  を乗じる速度を高める方法であって、

楕円曲線上のすべての点  $Q(x, y)$  について  $\psi(Q) = \lambda \cdot Q$  である自己準同形写像  $\psi$  が存在するように、有限体  $F$  の上の楕円曲線を選択するステップと、

スカラー  $k$  のより小さな表現  $k_i$  と写像  $\psi$  との組合せを使用して楕円曲線点  $Q$  のスカラー倍数を算出するステップとを含む方法が提供される。

#### 【0019】

本発明の好ましい実施形態のこれらおよび他の特徴は、添付の図面が参照される以下の詳細な説明でより明らかになろう。

#### 【0020】

##### 【発明の実施の形態】

以下の説明では、同じ符号は各図面中の同じ構造物を指す。図 1 を参照すると、データ通信システム 10 は、通信チャネル 16 によって接続され、送信側 12 および受信側 14 として指定された一対の通信者を含む。各通信者 12, 14 は、デジタル情報を処理し、この情報を後述のようにチャネル 16 を通して送信する準備をするこ

とのできる暗号化プロセッサ18、20をそれぞれ含む。プロセッサ18、20は、プロセッサに組み込まれた集積回路で実現するか、あるいは汎用プロセッサと共に所定のプロトコルを実施するようにデータキャリア上に符号化された命令として実現することができる。図をわかりやすくするために、通信者12は、比較的限られた計算力を持つ専用プロセッサ18を有するスマート・カードの形であると仮定する。好ましくは、プロセッサ20は、チャネル16によってカードと通信する中央サーバであり、チャネル16は無線通信チャネルである。

#### 【0021】

暗号化プロセッサ18は、ECCの楕円曲線暗号化システムを実現し、また暗号化プロセッサ18の機能の1つは、整数であるkと、基本楕円曲線上の点であるQとを暗号化方式における鍵対k、kQとして使用できるように、k・Qの形の点乗算を実行することである。上記で指摘したように、楕円曲線点とスカラー値の乗算などの暗号化計算はコストがかかる。

#### 【0022】

楕円曲線点Q(x, y)のスカラー乗算の速度を高める方法は、図2に示されており、全体的に符号50によって示されている。本発明のアルゴリズムは、プロセッサ12がたとえば、特定の種類の楕円曲線に関してメッセージに署名しメッセージを検証する速度を高める。この方法は、 $F_q$  ( $q$ は素数べき乗) として例示された有限体上の楕円曲線Eに関する以下の一般的な数式が与えられ、

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

自己準同形写像 $\psi$ が存在し、楕円曲線上のすべての点Q(x, y)点について $\psi(Q) = \lambda \cdot Q$ である場合、点Qと整数kとの乗算は、kのより小さな表現 $k_j$ と写像 $\psi$ との組合せを利用することによって速度を高めることができるという考えに基づく方法である。写像 $\psi$ は、 $kQ$ に関する後の計算で使用できる群要素およびそれらの組合せを事前に算出することも可能にする。

#### 【0023】

次に図2を参照すると、楕円曲線上の点乗算の速度を高める一般的な実施形態のフローチャートが符号50で示されている。まずシステム・パラメータが選択される。最初のステップとして、ある特性を有する基本楕円曲線Eが選択される。

本発明の第1の実施形態では、一般化された楕円曲線（1）を以下の形式で表すことができる。

$$E: y^2 = x^3 + b \pmod{p} \quad (P \text{ は素数}) \quad (2)$$

#### 【0024】

まず、 $\gamma \in F_p$  ( $F_p$  はすべての整数  $\pmod{p}$  から成るサイズ  $p$  の体である) であり、かつ  $\gamma^3 \equiv 1 \pmod{p}$  (1の立方根) である数  $\gamma$  が存在するように係数  $p$  を求めることができる。たとえば、 $p = 7$  である場合、 $2^3 \pmod{7} = 1$  であるので、 $\gamma = 2$  である。このような  $\gamma$  はすべての  $p$  に対して存在するわけではなく、したがって、 $p$  の値を選択する際にはこのことを考慮しなければならない。通常、適切な暗号強度を得るには、選択される  $p$  の長さが少なくとも 160 ビットであるべきである。

#### 【0025】

曲線  $E$  が選択された後、写像関数  $\psi$  が求められる。写像関数  $\psi: (x, y) \rightarrow (\gamma x, y)$  は、曲線上の点のある組を曲線上の点の別の組に写像するに過ぎない。楕円曲線  $E$  上の当該のすべての点  $Q(x, y)$  点について  $\psi(Q) = \lambda \cdot Q$  になるような整数  $\lambda$  が存在する。この整数  $\lambda$  は、 $\lambda^3 \equiv 1 \pmod{n}$  であることに留意することによって求めることができる。この場合、 $n$  は、 $F_p$  の上にある楕円曲線  $E$  上の点の数、すなわち、 $E(F_p)$  上の点の数である。 $\lambda^3 \equiv 1$  の  $\lambda$  に複数の解が存在する場合があるが、写像関数  $\psi$  を満たす解は 1 つだけである。 $\gamma^3 \pmod{p} = 1$  であるので、 $Q$  と  $\psi(Q)$  の両方が  $E$  に関する数式を満足する。したがって、長い計算を実行して  $\lambda$  による乗算の結果を求める代わりに、 $\lambda$  による乗算を非常に効率的に実行できるように写像関数の結果を使用して非常に効率的にこれを行うことができる。

#### 【0026】

シード点  $Q$  が選択され、52に示すように、製造時に、システム・パラメータ  $E$ 、 $p$ 、 $Q$ 、 $\lambda$ 、 $\psi(Q)$ 、および  $\gamma$  が、暗号化プロセッサ 18 によって使用できるよう記憶される。暗号化、鍵一致、署名などの暗号化手順を実施するには、短命秘密鍵  $k$  として使用される整数  $k$  を選択し、対応する公開鍵  $kQ$  を生成する必要がある。

#### 【0027】

$k$  の値は次式のように表すことができる。

$$k = (k_0 + k_1 \lambda) \bmod n \quad (3)$$

### 【0028】

上式で、 $n$  は  $E(F_p)$  上の点の数であり、 $k_0$  および  $k_1$  は整数である。この場合、点  $k \cdot Q$  は次式のようになる。

$$k \cdot Q = (k_0 Q + k_1 \lambda Q) \bmod n \quad (4)$$

### 【0029】

いくつかの暗号演算の場合、 $k$  の値をランダムに選択することができ、このような場合、 $k$  を選択するのではなく、符号ビットを含まない長さ  $\lceil \log_2(n) \rceil / 2$  の  $k_0$  および  $k_1$  の値をランダムに選択し（すなわち、 $k_1$  の長さとして、長さ  $k$  の少なくとも 2 分の 1 の長さが選択される）、次いで数式 (3) を使用して  $k$  の値を算出することが可能である。図 2 に 54 で示すように  $k_0$ 、 $k_1$  の値を選択した後、Me nezes らによって「Handbook of Applied Cryptography」(HAC) に記載された「Simultaneous Multiple Exponentiation」（アルゴリズム 14.88）に類似しており、56 に示されているアルゴリズムを使用して数式 (4) の右辺を高速に計算することができる。都合上、このアルゴリズムを以下に再現する。加群において、べき乗が加算に類似しており、したがって、このアルゴリズム内の乗算を加算で置き換えると以下の手順が得られる。

### 【0030】

#### アルゴリズム 1 同時多重加算

入力：群要素  $g_0, g_1, \dots, g_{|t|-1}$  および負でない  $t$  ビット整数  $e_0, e_1, \dots, e_{|t|-1}$ 。

出力： $g_0e_0 + g_1e_1 + \dots + g_{|t|-1}e_{|t|-1}$ 。

ステップ 1。事前計算。 $i$  が 0 から  $(2^l - 1)$  に対して、

#### 【数 2】

$$G_i \leftarrow \sum_{j=0}^{l-1} g_j i_j$$

上式で、 $i = (i_{|t|-1} \dots i_0)_2$  である。

ステップ 2。 $A \leftarrow 0$

ステップ 3。 $i$  が 1 から  $t$  に達するまで、以下のことを実行する。

$A \leftarrow A + A, A \leftarrow A + G_1$

ステップ4。 (A) を返す。この場合、 $A = g_0e_0 + g_1e_1 + \cdots + g_{|I|-1}e_{|I|-1}$

### 【0031】

このアルゴリズムを数式(4)に適用すると、2つの群要素 $g_0, g_1$ 、すなわち $Q$ および $\lambda Q$ があり、したがって、2つの整数 $e_0, e_1$ 、すなわち $k_0, k_1$ があることがわかる。このアルゴリズムでは、いくつかの値を事前に算出することができ、最初に $G_i$ が算出される。 $I = 2I$ によって $G_i$ を事前に算出した結果をテーブル1(表1)に示す。

### 【表1】

i	0	1	2	3
$G_i$	0	$g_0$	$g_1$	$g_0 + g_1$

### 【0032】

点加算： $(Q + \psi(Q))$  を実行して点を作成した後、算出済みの要素をテーブル1に記入してテーブル2(表2)を作成することが可能である。図2のステップ58に示すように、これらの要素を事前に算出しメモリに記憶することができる。

### 【表2】

i	0	1	2	3
$G_i$	0	$Q$	$\psi(Q)$	$Q + \psi(Q)$

$G_{|I|}$ を求め、したがって、60に示すように $|I_1$ から $|I_t|$ を求めないかぎり、アルゴリズムのステップを実行することはできない。 $k_i$ の2進表現を使用して概念行列または組合わせテーブルを作成することができる。たとえば、 $k_0 = 30$ であり $k_1 = 10$ である場合、 $k_0$ から $k_1$ の2進表現内の最大ビット数が5であるので $t$ は値5を有する。 $k_0$ から $k_1$ の2進表現で作成された概念行列をテーブル3に示す。 $|I_i|$ は、第1の行が最下位ビットを含み、第2の行が次の下位ビットを含み、以下同様である $i$ 番目の列に表された数によって決定される。したがって、テーブル3(表3)から、 $|I_1| = |I_2| = (11) = 3, |I_3| = (01) = 1, |I_4| = 3$ 、および $|I_5| = 0$ であることがわかる。

【表3】

i	1	2	3	4	5
k <sub>0</sub>	1	1	1	1	0
k <sub>1</sub>	0	1	0	1	0
I <sub>i</sub>	1	3	1	3	0

## 【0033】

このアルゴリズムを完了するのに必要なすべての成分を得ることができ、62に示すようにステップ3の反復が行われる。

## 【0034】

最初はA←0であり、iは1に設定される。I<sub>i</sub> = I<sub>1</sub>であり、これはテーブル3から1に等しい。したがって、G<sub>1|1</sub>はG<sub>1</sub>であり、これはテーブル2からQである。したがって、I = 1の場合の反復によるAの値は0 + Q = Qである。

## 【0035】

i = 2である次の反復の場合、Aの初期値はQであり、したがって、A ← Q+Q = 2Qであり、テーブル3からI<sub>i</sub> = I<sub>2</sub> = 3である。したがって、G<sub>1|2</sub>はテーブル3からG<sub>3</sub>に等しく、すなわち、Q + ψ(Q)である。

## 【0036】

したがって、A + G<sub>1|i</sub>は2Q + Q + ψQ = 3Q + ψQと算出される。

## 【0037】

反復は、5回目の反復まで、すなわち、k<sub>0</sub>q = k, λQの値、すなわちkQが算出されるまで、テーブル4（表4）に記載されたiの値ごとに継続される。

【表4】

i	A
1	Q
2	3 Q + ψ (Q)
3	7 Q + 2 ψ (Q)
4	15 Q + 5 ψ (Q)
5	30 Q + 10 ψ (Q)

## 【0038】

各反復では、点倍加 (A+A) および点加算 (A+G<sub>1|i</sub>) が必要である。ただし、場

合によっては、 $G_{|i}$ の値は〇であり、この場合、計算量が削減される。

#### 【0039】

したがって、この方法では、 $\max\{\log_2(k_i)\}$ に等しい数の点倍加と、ほぼ同数の点加算が必要であることがわかる。点加算の数は、ウィンドウ技法（アルゴリズム14.85 HAC）および指数再符号化技法を使用して削減することができる。 $i$ および $G_i$ の値を事前に算出することができるので、事前に算出された適切な要素 $G_i$ をテーブル2から検索することによって点加算を容易に実行することができる。kPが算出された後、チャネル16を介した暗号化伝送または署名伝送において通信者12の短命公開鍵としてこのkPを使用することができる。

#### 【0040】

簡単に言えば、暗号化やDH署名などの暗号化演算の場合、整数 k が必要であり、対応する公開鍵 kQ が算出される。それぞれ、長さ n の 2 分の 1 の長さを有する、値  $k_0$  および  $k_1$  がランダムに選択され、適切なアルゴリズムを使用して項  $k_0 Q = k_1 \lambda Q$  が生成される。k をこのように選択すると、この方法は、k 自体をランダムに生成するのと同程度に安全であると思われる。もちろん、効率を向上するためにより少ないビット数の  $k_i$  を選択することが可能である。

#### 【0041】

上記の技法において、 $k = k_0 + k_1 \lambda$  を書き込み、同時に組み合わせる方法によって、同時多重加算アルゴリズムの速度が高められる。 $k = k_0 + k_1 \lambda$  を書き込む技法は、スカラーベンチ技法、すなわち、ワインディング、組合せなどと共に使用することも有利である。

#### 【0042】

いくつかの写像  $\psi$  では、3つ以上の下位要素  $k$  (sub k) を使用することも可能である。いくつかの写像  $\psi$  では、 $k = k_0 + k_1 \lambda + k_2 \lambda^2$  を書き込み、同時多重加算アルゴリズムを適用することによって k の値を算出することができる。

#### 【0043】

本発明の第2の実施形態では、一般化された橢円曲線式(1)の異なる形式、すなわち、

$$y^2 = (x^3 - ax) \bmod p \quad (5)$$

が使用される。この場合も、 $p$ は少なくとも160ビットを有する素数である。この種の曲線の場合、 $\gamma$ に必要な特性が異なる。この場合、 $\gamma^2 \equiv -1 \pmod{p}$ になるような値を求める必要がある。 $\gamma$ の特性を変更するには、異なる写像関数 $\psi'$ を使用する必要がある。この実施形態では、写像は形式 $\psi' : (x, y) \rightarrow (-x, \gamma y)$ をとる。 $(x, y)$ が曲線上にある場合は $\psi'(x, y)$ も曲線上にある。この場合、 $\lambda^4 \equiv 1 \pmod{n}$  ( $n$ はこの場合も、 $E(\mathbb{F}_p)$ 上の点である)、したがって $\lambda$ を算出することができる。写像 $\psi'(Q) = \lambda \cdot Q$ は前述のように実行され、この場合も、この曲線に対して $\lambda$ による乗算を非常に効率的に行うことができる。この実施形態の $k$ に関する式は、第1の実施形態と同じであり、次式によって表される。

$$k = (k_0 + k_1 \lambda) \pmod{n} \quad (6)$$

この式は、前の実施形態と同じであり、2つの群要素のみを有する。したがって、アルゴリズム1の群要素 $Q$ および $Q + \psi'(Q)$ を使用して点 $k \cdot Q$ を算出することができる。この計算では、 $\max\{\log_2(k_j)\}$ に等しい数の点倍加と、同様な数の点加算が必要である。前述のように、ウィンドウ技法および指数再符号化技法を使用して点加算の数を削減することができる。

#### 【0044】

この方法は、効率的に算出できる自己準同形写像 $\psi$ が存在するかぎり、他の橢円曲線にも適用することができる。

#### 【0045】

上記の実施形態は、 $k$ をランダムに選択することができ、したがって、 $k_0$ および $k_1$ を選択することができるものと仮定し、 $k$ を求めている。 $k$ を選択することができる可能な暗号化プロトコルの場合、まず、 $k = (k_0 + k_1 \lambda) \pmod{n}$ になるように $k$ の所与の値から所望の「短い」形式の $k_0, k_1$ を求める必要がある。場合によつては、3つ以上の $k$ を使用することができ有利である。

#### 【0046】

上述の実施形態でわかるように、事前に点がわかっているときは、テーブルを作成して乗算の速度を高めることができる。しかし、未知の点の倍数が必要になる場合があり（たとえば、これはECDSA検証で起こる可能性がある）、その場合

、与えられた  $k$  の値をとり、次いで  $k_i$  の適切な表現を決定する必要がある。

#### 【0047】

したがって、第3の実施形態では、システム・パラメータおよび値  $k$  が与えられ、点  $Q$ 、必要な倍数  $k$ 、および複素乗算倍数  $\lambda$  が既知である。所定の  $k$  の値から「短い」  $k_i$  を求める必要がある。これを行う方法を以下に説明し、図3のフローチャートに示す。（ $k$  を必要としない）事前計算として、 $a_i$  および  $b_i$  が  $n$  よりも小さな数になるような2つの関係を算出する。

$$a_0 + b_0 \lambda = 0 \bmod n$$

$$a_1 + b_1 \lambda = 0 \bmod n$$

$a_i$  および  $b_i$  はできるだけ小さいことが好ましいが、本発明の方法は、 $a_i$  および  $b_i$  が最小限でないときでも有利である。 $a_i$  および  $b_i$  が共に小さな対  $a_i$  および  $b_i$  は、小さなユークリッド長を有するベクトル  $u_i$  とみなすことができる。通常、後述の方法では、最初の  $k$  のサイズの2分の1の表現を有する  $k_0$  および  $k_1$  が生成される。

#### 【0048】

本実施形態では、事前に算出された短いベクトル表現を使用して以下の形式の式を得ることによって、 $kQ$  を効率的に算出することができる。

$$k_0 Q + \lambda k_1 Q$$

#### 【0049】

これは、事前に算出されたベクトルを使用して、 $k$  の知識を必要としない分数  $f_0$  および  $f_1$  を導くことによって行われる。ベクトル  $z$  は、分数  $f_0$  および  $f_1$  と  $k$  を組み合わせることによって生成される。ベクトル  $z$  を使用して、 $v' = (v_0', v_1')$  である第2のベクトル  $v'$  が算出され、 $kQ$  の値が、

#### 【数3】

$$v_0' Q + \lambda v_1' Q \quad (8)$$

として算出される。この解を得る方法を以下に詳しく説明する。

#### 【0050】

小さな  $a_i$  および  $b_i$  を作成する場合、 $L_3$ 、すなわち、短い基底ベクトルが直接得

られる格子基底削減アルゴリズム (HAC、118ページ) を利用することが可能である。しかし、この好ましい実施形態では、簡単な拡張ユークリッド・アルゴリズムが対  $(n, \lambda)$  に対して使用される。 $(n, \lambda)$  に対する拡張ユークリッド・アルゴリズムにより、 $i$  に応じて  $r_i$  の表現 (たとえば、ビット長) が小さくなり、 $c_i$  および  $d_i$  の表現が大きくなる線形組合せ  $c_i n + d_i \lambda = r_i$  が作成される。

#### 【0051】

拡張ユークリッド・アルゴリズムを使用した結果として得られた  $| (d_i, r_i) |$  の 2 つの最小値が保存される。これらのベクトルのサイズが、平方ユークリッド基準  $| (d_i, r_i) | = d_i^2 + r_i^2$  を用いて測定される。これらの最小関係の項は、

$$\hat{d}_0, \hat{r}_0$$

および

$$\hat{d}_1, \hat{r}_1$$

として示され、通常、アルゴリズムの中央で得られる。最小関係が保持されない場合でも、この方法は、部分最適関係により、依然として点倍数の計算において有利である。

#### 【0052】

$a_i$  および  $b_i$  の値は、すべて事前に算出することのできる

$$a_0 = -\hat{r}_0, b_0 = \hat{d}_0$$

および

$$a_1 = -\hat{r}_1, b_1 = \hat{d}_0$$

を定義することによって作成することができる。

#### 【0053】

次のタスクは、倍数  $k$  の小さな表現を求めることがある。

#### 【0054】

$a_0, b_0$  および  $a_1, b_1$  の計算が与えられた場合、 $u_0 = (a_0, b_0)$  であり  $u_1 = (a_1,$

$b_1)$ であるベクトル $u_0, u_1$ を指定することが可能である。これらのベクトルは $a_i + b_i \lambda = 0 \pmod{n}$ を満たす。群要素Qにベクトル $v = (v_0, v_1)$ を乗じることは $(v_0 + v_1 \lambda)Q$ として定義される。 $a_i + b_i \lambda = 0 \pmod{n}$ であるので、任意の群要素Rについて $u_0R = u_1R = 0$ が成立する。したがって、任意の整数 $z_0$ および $z_1$ の場合、任意の群要素Rについて $v' R = (v - z_0u_0 - z_1u_1)R$ が成立する。

#### 【0055】

整数 $z_0$ および $z_1$ としては、ベクトル $v' = v - z_0u_0 - z_1u_1$ ができるだけ小さな成分を有するような整数が選択される。この場合も、この方法は、 $v'$ の成分が小さい場合に利点を有し、該成分が必ずしも最小限でない場合でも利点を有する。

#### 【0056】

適切な $z_0$ および $z_1$ は、 $v$ の基底を正規直交基底 $\{u_0, u_1\}$ に変換することによって算出される。基底間の変換では行列乗算が行われる。ベクトル $v = (v_0, v_1)$ を $\{u_0, u_1\}$ 基底から標準基底 $\{(1, 0), (0, 1)\}$ に変換する場合、次式が成立する。

#### 【数4】

$$v_{\{(1,0),(0,1)\}} = v_{(u_0, u_1)} M = (v_0, v_1) \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$$

他方の方向、すなわち、正規直交基底 $\{(1, 0), (0, 1)\}$ から $\{u_0, u_1\}$ 基底に変換する場合、乗算は単にMの逆数によって行われる。

#### 【数5】

$$v_{(u_0, u_1)} = v_{\{(1,0),(0,1)\}} \text{inverse}(M) = v_{\{(1,0),(0,1)\}} \frac{1}{a_0 b_1 - a_1 b_0} \begin{bmatrix} b_1 & -b_0 \\ -a_1 & a_0 \end{bmatrix}$$

#### 【0057】

ベクトル $v = (k, 0)$ がゼロ成分を有するので、逆数(M)の下の行は必要とされず、したがって、 $\{u_0, u_1\}$ に変換する場合に必要なのは次式の分数だけである。

#### 【数6】

$$f_0 = \frac{b_1}{a_0 b_1 - a_1 b_0}$$

および

**【数7】**

$$f_1 = \frac{b_0}{a_0 b_1 - a_1 b_0}$$

**【0058】**

分数 $f_0$ および $f_1$ の演算を乗算のみで行うことができるよう、これらの分数を十分な精度に事前に算出しておくことができる。これらの分数をもたらす計算が $k$ に依存せず、したがって、楕円曲線がシステム・パラメータとして選択される際にこれらの分数を1度算出することができ、各 $k$ ごとに再計算する必要はない。同様に、ベクトル $v$ 、 $u_0$ 、および $u_1$ を事前に算出し記憶することができる。

**【0059】**

$k$ の値が選択されるか、あるいは求められた後、まず、 $z$ が $(z_0, z_1) = (\text{round}(kf_0), \text{round}(kf_1))$ として定義される $z = (z_0, z_1)$ を計算することによって $kQ$ の値を算出することができる。 $z$ の近傍の他のベクトルも有用であり、したがって、丸めを床関数または天井関数、あるいは何らかの他の近似で置き換えることができる。

**【0060】**

適切な $z$ が求められた後、 $v' = (v_0', v_1') = v - z_0 u_0 - z_1 u_1$ によって $v(k, 0)$ の効率的な等価物が算出される。「効率的な等価物」の句は、 $v' P = vP$ および $v'$ が小さな係数を有するようなベクトル $v'$ を意味する。この場合、値 $kQ$ は $v_0' Q + v_1' \lambda Q$ として算出される。この値は、上述のように同時点加算を使用して算出することができ、上記で説明し、かつH.A.C.14.7の627ページに記載されているように非隣接形式(NAF)再符号化を使用することによってより高い効率を得ることができる。したがって、 $k$ が所定の値である場合でも、 $k_0$ および $k_1$ の値を算出し写像関数と共に使用して $kQ$ の値、したがって、鍵対 $k$ 、 $kQ$ を得ることができる。

## 【0061】

$k$  を 3 つの部分  $k = k_0 + k_1 \lambda + k_2 \lambda^2$  に 分離する場合、 $L^3$  行削減によって、次式のような小さなベクトルを得ることができる。

## 【数8】

$$\begin{bmatrix} 1 & 0 & -\lambda^2 \\ 0 & 1 & -\lambda \\ 0 & 0 & -n \end{bmatrix} \text{ to } \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$$

## 【0062】

二次元の場合と同様に小ベクトル等価物（三次元行）を得ることができる。

## 【0063】

これら の方法を使用して  $k \cdot Q$  の値を求めるとき、暗号化プロセッサ12によって必要とされる処理能力が大幅に低減される。また、このような反復計算が行われる速度が高められ、それによって、情報を転送するための時間が短縮される。

## 【0064】

当然のことながら、スカラー倍数  $k$  が短縮成分  $k = k_0 + k_1 \lambda + k_2 \lambda^2 + \dots + k_{m-1} \lambda^{m-1}$  で表された後、同時多重加算アルゴリズムの代わりに、あるいはこのアルゴリズムと共に、効率的な楕円曲線スカラー乗算を行うための他の方法を使用することができる。このような方法には、ウインドウ技法（固定およびスライド）、組合せ技法、ピット再符号化技法、およびこれらの技法の組合せが含まれる。

## 【0065】

特に有益なある技法では、乗算の一成分、たとえば  $k_0$  に関して作成されたテーブルを他の成分  $k_1$  などに再使用することができる。これは、必要に応じて写像  $\gamma$  を適用することにより、算出されたテーブル要素を変換することによって行われる。

## 【0066】

他の例として、 $k$  を  $k = k_0 + k_1 \lambda + k_2 \lambda^2$  として再設定することができ、 $k$  が  $m$  ピットを有し、 $k_1$  が約  $m/3$  ピットを有する実施形態について以下に説明する

## 【0067】

成分 $k_j$ は、求められた後、2進表現から、より少ない非ゼロ・ビットを有する符号付き2進表現に再符号化することができる。この再符号化は、表現 $k_j$ 内のある1ビットまたは-1ビットが符号付き2進文字列内で他の非ゼロに隣接しないような非隣接形式(NAF)をとることができる。

## 【0068】

各 $k_j$ が再符号化された後、 $k_j \lambda^i P$ の計算を助けるためにテーブルを作成することができる。

## 【0069】

NAFウィンドウ・テーブルは、 $\lambda^i P$ のある短ビット長倍数を事前に算出するためのテーブルである。このウィンドウの幅によってテーブルのサイズが決定される。 $k_j$ が、隣接する非ゼロを有さないように再符号化されているので、奇数ウィンドウ幅が適切である。3ビット幅NAFウィンドウは以下のものを含む。

## 【数9】

1	1 0 1	1 0 - 1
---	-------	---------

## 【0070】

再符号化 $k_j$ は、このようなウィンドウを連結し、必要に応じてゼロを充填することによって作成される(H.A.C.、616ページ)。

## 【0071】

あらゆる非ゼロ・ビットではなく、発生するあらゆるウィンドウごとにのみEC点を加算または減算するだけでよいので、このテーブルを使用することによって必要な加算数を削減することができる。

## 【0072】

したがって、まず、この技法は $k_0 P$ の計算に適用される。

## 【0073】

$k_0 P$ の計算のために作成されたテーブルは、演算子 $\gamma$ を使用してテーブル要素

に $\psi$ 写像が写像される場合に $k_1 \lambda P$ 計算に適用することができる。同様に、 $k_0 P$ 用に作成されたテーブルを使用するが、テーブル要素に $\gamma^2$ を写像することによって、 $k_2 \lambda^2 P$ の速度を高めることができる。

#### 【0074】

スライド・ウィンドウ技法を該成分に適用する際に、実行する必要があるのは1組の倍加だけである。

#### 【0075】

好ましい実施形態のこの例を示すために、以下の例を使用する。

#### 【0076】

$k = [101101011101]_2 + [111010101101]_2 \lambda$  である場合、

以下の再符号化を行う。

$$k = [10-100-10-100-101] + [1000-10-10-10-101] \lambda$$

$$= k'_0 + k'_1 \lambda$$

#### 【0077】

$1 \cdot P$ 、 $[10-1]$ ・ $P$ 、 $[101]$ ・ $P$ を含む、 $P$ に関する3ビット・ウィンドウ・テーブルが事前に算出される。この場合、2回のEC加算および2回のEC倍加が必要である。

#### 【0078】

この後、テーブルの要素を加算／減算することによって、 $kP$ を以下のように算出することができる。

$$kP = [10-100-10-100-101]P + [1000-10-10-10-101] \cdot \lambda P$$

これは、累算器Aを使用して以下のように行うことができる。

$A \leftarrow 0$  ; 初期設定

$A += \psi(1 \cdot P)$  ;  $k_0'$  の上位ビットを消費する。

$A \leftarrow 2A$  ; Aを倍加する。

$A \leftarrow 2A$

$A \leftarrow [10-1]P$  ;  $k_0'$  の上位3ビットを消費する。

$A \leftarrow 2^4A$

$A -= [101]\psi P$  ;  $k_1'$  の3ビット・ウィンドウを消費する。

$A \leftarrow 2A$  ; Aを倍加する。

$A^- = [101]P$  ;  $k_1'$  の3ビットを消費する。

$A \leftarrow 2^4A$

$A^- = [101]\psi P$  ;  $k_1'$  の3ビットを消費する。

$A \leftarrow 2^2A$

$A^- = [10-1]P$  ;  $k_0'$  の最後のビットを消費する。

$A^+ = \psi P$  ;  $kP$ を生成する。

#### 【0079】

簡単に言えば、前述の技法は以下のとおりである。橢円曲線Eおよび自己準同形写像 $\psi$ が与えられた場合、すべての点 $Q \in E$ について $\lambda Q = \psi(Q)$ になるような対応する整数 $\lambda$ が存在する。整数 $m$ を選択し、同等な数 $m$ の「短基底ベクトル」 $b_1, b_2, \dots, b_m$ を算出する。このような基底ベクトルはそれぞれ整数に対応し、そのような整数はそれぞれ、点の数 $n = \#E(F_p^m)$ （すなわち、点の数）で除ることができる。次に、整数 $k$  ( $0 < k < n$ ) が与えられた場合、 $k_i$ として「短い」ベクトルが選択される $k = \sum k_i \cdot \lambda^i$ と書くことができる。これは、 $b_1, b_2, \dots, b_m$ によって生成された格子内の（ $k$ を表す）あるベクトルと近傍のベクトルとの差を求ることによって行われる。

#### 【0080】

以下の実施形態では、複合体の上に画定された橢円曲線に前述の技法（自己準同形写像および基底変換および「Shamir's trick」）を適用することについて明白に説明する。特に、 $p$ が奇素数である曲線 $E(F_p^m)$ への適用について説明する。以下の実施形態ではこのような曲線に対する技法を例示する。

#### 【0081】

この技法について、写像 $\psi$ がフロベニウス写像 $\psi(x, y) = (xp, yp)$ であり、 $A, B \in F_p$ である $E' A, B(F_p^m)$ が使用される場合において説明する。

#### 【0082】

この場合、フロベニウス写像が $\psi^2 - t\psi + p = 0$ を満たし、 $t = p + 1 - \#E(F_p^m)$ であることがわかっている。

#### 【0083】

したがって、 $\lambda^2 - t\lambda + p = 0 \bmod n$ であり、また $\lambda^{2+i} - t\lambda^{1+i} + p\lambda^i = 0 \bmod n$ である。

[0084]

以下のベクトルが、ベクトル空間 $Q_n$ 空間の $m$ 個の「短」基底ベクトルで構成されている。

【数 10】

	$(\lambda^{m-1}, \dots)$	$\lambda^2, \lambda^1, \lambda^0)$
$b_1$	$(0, 0, 0, \dots)$	$0, 1, -t, p)$
$b_2$	$($	$1, -t, p, 0)$
	$(1, -t, p, 0, 0, \dots)$	$\dots, 0)$
	$(-t, p, 0, 0, \dots)$	$\dots, 0, 1)$
$b_m$	$(p, 0, 0, 0, \dots)$	$0, 1, -t)$

したがって、このような曲線に対する  $k \cdot Q$  を計算する場合、ベクトル  $b_1, b_2, \dots, b_m$  および前述の技法を使用することができる。

[0085]

上記の実施形態では、 $k$ 、 $\lambda Q$ を $\psi(kQ)$ から得ることができ、写像が加算よりも効率的であることが理解されよう。

[0086]

本発明を特定の実施形態を参照して説明したが、当業者には、添付の請求の範囲に記載された本発明の趣旨および範囲から逸脱しない本発明の様々な修正形態が明らかになろう。

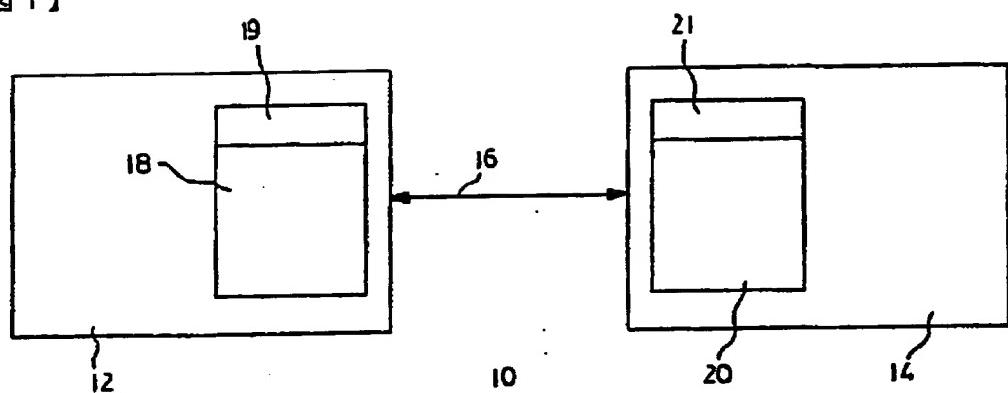
### 【図面の簡単な説明】

【図1】 通信システムの概略図である。

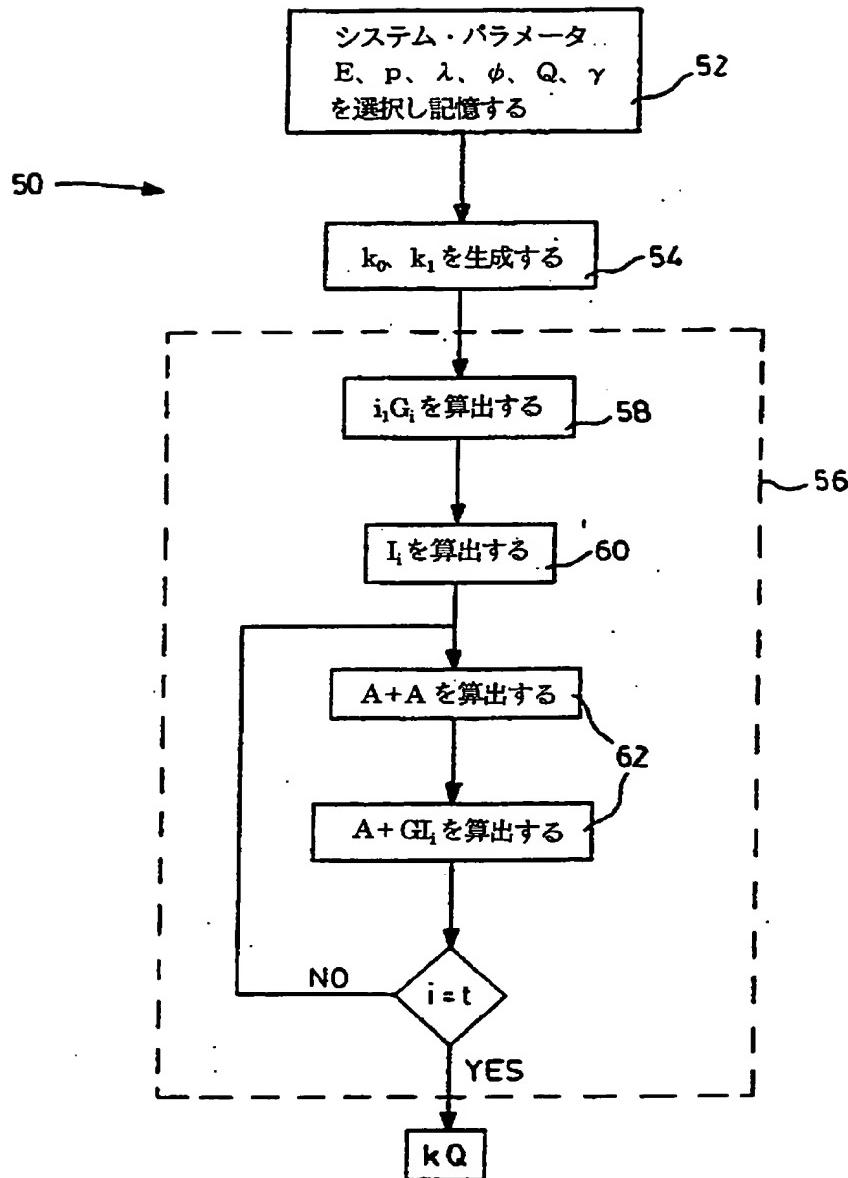
【図2】 本発明の第1の実施形態を実施する各ステップを示すフローチャートである。

【図3】 図2の方法を実施するために必要なパラメータを与えるステップを示すフローチャートである。

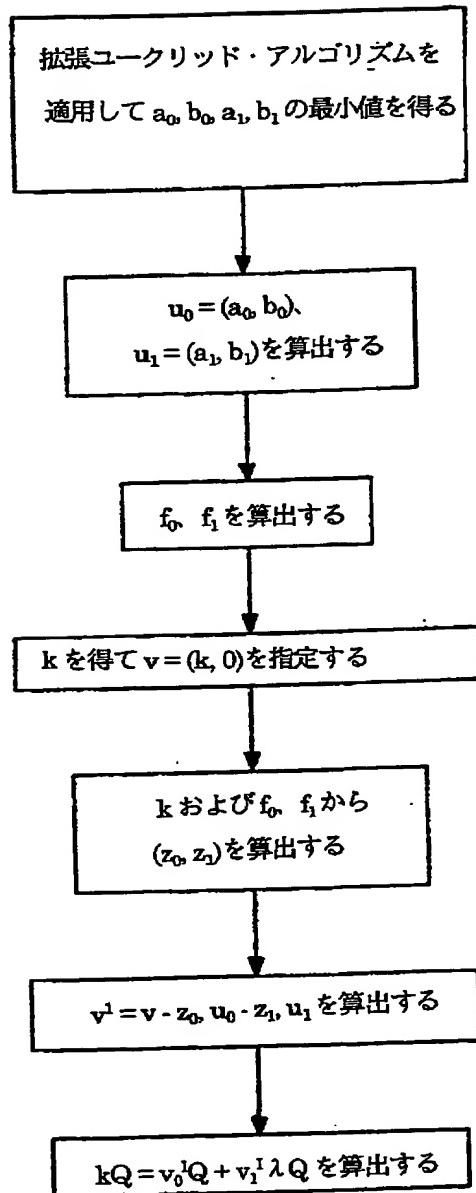
【図1】



【図2】



【図3】



**【手続補正書】**

【提出日】平成13年8月10日(2001.8.10)

**【手続補正1】**

【補正対象書類名】明細書

【補正対象項目名】発明の詳細な説明

【補正方法】変更

**【補正の内容】****【発明の詳細な説明】****【○○○1】****【発明の属する技術分野】**

本発明は、梢円曲線を利用して暗号システムで計算を実行する方法に関する。

**【○○○2】****【従来の技術】**

公開鍵データ通信システムを使用して一対の通信者の間で情報を転送することができる。交換される情報の少なくとも一部は、送信側により所定の演算によって暗号化され、受信側は、送信側の演算に対して相補的な演算を実行してこの情報を復号することができる。

**【○○○3】**

各通信者は、秘密鍵と、秘密鍵に数学的に関係づけられた公開鍵とを有する。この関係は、公開鍵の知識から秘密鍵を特定することができないような関係である。各鍵は、転送すべきデータを暗号化するために、あるいはデータが真正であることを検証できるように署名を添付するために、データの転送時に使用される。

**【○○○4】**

暗号化の場合、一方の通信者は、受信側の公開鍵を使用してメッセージを暗号化し、受信側に送信する。次いで、受信側は秘密鍵を使用してメッセージを復号する。

**【○○○5】**

一方の当事者の公開鍵を他方の当事者の秘密鍵と組み合わせることによって共

通鍵を生成することもできる。このような場合には通常、各当事者の長期鍵が破られるのを回避するため、通信セッションごとに新しい秘密鍵およびこれに対応する公開鍵が生成され、これらの鍵は通常、セッション鍵または短命鍵と呼ばれる。

#### 【0006】

したがって、メッセージの交換および公開鍵の生成では、暗号化システム $Z^*_p$ において整数mod p (pは素数) の有限体を利用する際に指数演算を伴い、あるいはシステムが橍円曲線を利用する際に同様な点乗算演算を伴う相当量の計算が行われる。橍円曲線システムでは、秘密整数 k を生成し、シード点 Q で点乗算を実行して短命公開鍵  $kQ$  を形成することによって、短命鍵対が得られる。同様に、共通短命セッション鍵を生成する場合は、公開鍵  $k_3Q$ 、すなわち、曲線上の点に他方の通信者の秘密整数  $k_b$  を乗じる必要があり、したがって、この場合も点乗算が必要になる。

#### 【0007】

メッセージに署名する場合にも、送信側が自分の秘密鍵をメッセージに適用することを除いて、同様な手順が使用される。この場合、任意の受信側が、送信側の公開鍵を使用してメッセージを復元し検証することができる。

#### 【0008】

このような方式を実施するための様々なプロトコルが存在しており、そのうちのいくつかは広く使用されている。しかし、それぞれの場合に、送信側は転送すべき情報に署名するために計算を実行する必要があり、受信側は、署名された情報を検証するために計算を実行する必要がある。

#### 【0009】

典型的な実行形式において、署名成分は以下の形式を有する。

$$s = ae + k \pmod{n}$$

上式で、橍円曲線暗号化システムにおいて、

P は、基本曲線上の、システムの定義済みパラメータである点であり、

k は、短期秘密鍵またはセッション鍵として選択されたランダム整数であり、

R = kP は、これに対応する短期公開鍵であり、

aは、送信側の長期秘密鍵であり、

$Q = aP$ は、これに対応する送信側の公開鍵であり、

eは、メッセージmおよび短期公開鍵Rの、SHA-1ハッシュ関数などの安全ハッシュであり、および

nは曲線の次数である。

#### 【0010】

送信側は、m、s、およびRを含むメッセージを受信側に送信し、署名は、Rに対応すべき値 $R^1 = (sP - eQ)$ を算出することによって検証される。算出された値が対応する場合、署名は検証されたことになる。

#### 【発明が解決しようとする課題】

#### 【0011】

検証を実行する場合、点乗算を計算し、それぞれ計算が複雑なsPおよびeQを得る必要がある。この場合、受信側が適切な計算力を有する場合には、特に問題はないが、セキュアトークン・アプリケーションや「スマート・カード」アプリケーションのように受信側の計算力が限られている場合、このような計算によって検証プロセスに遅延が生じる。

#### 【0012】

したがって、キー生成プロトコルおよび署名プロトコルは、大量の計算を必要とする場合がある。暗号化が普及するにつれて、より高速であり、スマート・カードや無線装置に見られるように限られた計算力を活用する暗号化システムを実現する要求が高まっている。

#### 【0013】

楕円曲線暗号化（ECC）は、この計算問題を解決する。ECCでは、鍵および証明書のサイズを縮小することができ、それによって、必要なメモリが低減し、コストが著しく節約される。ECCは、コストを著しく削減することができるだけでなく、次世代アプリケーションにおけるスマート・カードの普及を促進する。また、ECCアルゴリズムによって鍵サイズを縮小することができるが、より大きな鍵を用いる他のアルゴリズムと同じレベルのセキュリティが維持される。

#### 【0014】

しかし、暗号化装置の低い生産コストを維持しながら情報転送速度を高めるために、依然として、鍵に対する計算をより高速に行う必要がある。

#### 【0015】

橭円曲線上の点の倍数を算出することは、橭円曲線暗号化で最も頻繁に実行される計算の1つである。このような計算の速度を高める1つの方法は、事前計算された点の倍数のテーブルを使用することである。この技法は、点が事前にわかっているときにさらに有用である。しかし、未知の点の倍数が必要になる場合がある（たとえば、ECDSA検証）。したがって、点乗算を容易にするシステムおよび方法が必要である。

#### 【0016】

##### 【課題を解決するための手段】

一般に、本発明は、スカラー  $k$  を、成分  $k_i$  と、基本曲線で自己準同形写像から導かれる整数  $\lambda$  との組合せとして表す。

#### 【0017】

この方法は、有限体の上に写像される複素乗算を有する橭円曲線 (EC) が与えられた場合、複素乗算写像と点  $Q$  に  $\lambda$  を乗じることが等しくなる二次方程式の解  $\lambda$  が存在する、という考えに基づく方法である。 $\lambda$  を整数とみなして EC 乗算を実行するのと比べて、複素乗算写像を介して  $\lambda Q$  を算出する方がコストが低いことが多い。実際には、他のスカラー ( $\lambda$  以外) による点乗算が必要になる。乗算写像を使用して点の他の倍数を算出できることも示す。

#### 【0018】

本発明によれば、橭円曲線点  $Q(x, y)$  にスカラー  $k$  を乗じる速度を高める方法であって、

橭円曲線上のすべての点  $Q(x, y)$  について  $\psi(Q) = \lambda \cdot Q$  である自己準同形写像  $\psi$  が存在するように、有限体  $F$  の上の橭円曲線を選択するステップと、スカラー  $k$  のより小さな表現  $k_i$  と写像  $\psi$  との組合せを使用して橭円曲線点  $Q$  のスカラー倍数を算出するステップとを含む方法が提供される。

#### 【0019】

本発明の好ましい実施形態のこれらおよび他の特徴は、添付の図面が参照され

る以下の詳細な説明でより明らかになろう。

### 【0020】

#### 【発明の実施の形態】

以下の説明では、同じ符号は各図面中の同じ構造物を指す。図1を参照すると、データ通信システム10は、通信チャネル16によって接続され、送信側12および受信側14として指定された一対の通信者を含む。各通信者12、14は、デジタル情報を処理し、この情報を後述のようにチャネル16を通して送信する準備をすることのできる暗号化プロセッサ18、20をそれぞれ含む。プロセッサ18、20は、プロセッサに組み込まれた集積回路で実現するか、あるいは汎用プロセッサと共に所定のプロトコルを実施するようにデータキャリア上に符号化された命令として実現することができる。図をわかりやすくするために、通信者12は、比較的限られた計算力を持つ専用プロセッサ18を有するスマート・カードの形であると仮定する。好ましくは、プロセッサ20は、チャネル16によってカードと通信する中央サーバであり、チャネル16は無線通信チャネルである。

### 【0021】

暗号化プロセッサ18は、ECCの楕円曲線暗号化システムを実現し、また暗号化プロセッサ18の機能の1つは、整数であるkと、基本楕円曲線上の点であるQとを暗号化方式における鍵対k、kQとして使用できるように、k・Qの形の点乗算を実行することである。上記で指摘したように、楕円曲線点とスカラ一値の乗算などの暗号化計算はコストがかかる。

### 【0022】

楕円曲線点Q(x, y)のスカラ一乗算の速度を高める方法は、図2に示されており、全体的に符号50によって示されている。本発明のアルゴリズムは、プロセッサ12がたとえば、特定の種類の楕円曲線に関してメッセージに署名しメッセージを検証する速度を高める。この方法は、 $F_q$  ( $q$ は素数べき乗) として例示された有限体上の楕円曲線Eに関する以下の一般的な数式が与えられ、

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

自己準同形写像 $\psi$ が存在し、楕円曲線上のすべての点Q(x, y)点について $\psi(Q) = \lambda \cdot Q$ である場合、点Qと整数kとの乗算は、kのより小さな表現 $k_i$ と写像 $\psi$ と

の組合せを利用してすることによって速度を高めることができるという考えに基づく方法である。写像 $\psi$ は、 $kQ$ に関する後の計算で使用できる群要素およびそれらの組合せを事前に算出することも可能にする。

#### 【0023】

次に図2を参照すると、橭円曲線上の点乗算の速度を高める一般的な実施形態のフローチャートが符号50で示されている。まずシステム・パラメータが選択される。最初のステップとして、ある特性を有する基本橭円曲線Eが選択される。本発明の第1の実施形態では、一般化された橭円曲線(1)を以下の形式で表すことができる。

$$E: y^2 = x^3 + b \pmod{p} \quad (P \text{ は素数}) \quad (2)$$

#### 【0024】

まず、 $\gamma \in F_p$  ( $F_p$ はすべての整数 $\pmod{p}$ から成るサイズ $p$ の体である) であり、かつ $\gamma^3 \equiv 1 \pmod{p}$  (1の立方根) である数 $\gamma$ が存在するように係数 $p$ を求めることができる。たとえば、 $p = 7$ である場合、 $2^3 \pmod{7} = 1$ であるので、 $\gamma = 2$ である。このような $\gamma$ はすべての $p$ に対して存在するわけではなく、したがって、 $p$ の値を選択する際にはこのことを考慮しなければならない。通常、適切な暗号強度を得るには、選択される $p$ の長さが少なくとも160ビットであるべきである。

#### 【0025】

曲線Eが選択された後、写像関数 $\psi$ が求められる。写像関数 $\psi: (x, y) \rightarrow (\gamma x, y)$ は、曲線上の点のある組を曲線上の点の別の組に写像するに過ぎない。橭円曲線E上の当該のすべての点 $Q(x, y)$ 点について $\psi(Q) = \lambda \cdot Q$ になるような整数 $\lambda$ が存在する。この整数 $\lambda$ は、 $\lambda^3 \equiv 1 \pmod{p}$ であることに留意することによって求めることができる。この場合、 $n$ は、 $F_p$ の上にある橭円曲線E上の点の数、すなわち、 $E(F_p)$ 上の点の数である。 $\lambda^3 \equiv 1$ の $\lambda$ に複数の解が存在する場合があるが、写像関数 $\psi$ を満たす解は1つだけである。 $\gamma^3 \pmod{p} = 1$ であるので、 $Q$ と $\psi(Q)$ の両方がEに関する数式を満足する。したがって、長い計算を実行して $\lambda$ による乗算の結果を求める代わりに、 $\lambda$ による乗算を非常に効率的に実行できるように写像関数の結果を使用して非常に効率的にこれを行うことができる。

## 【0026】

シード点Qが選択され、52に示すように、製造時に、システム・パラメータE、p、Q、λ、ψ(Q)、およびγが、暗号化プロセッサ18によって使用できるようカーデ12に記憶される。暗号化、鍵一致、署名などの暗号化手順を実施するには、短命秘密鍵kとして使用される整数kを選択し、対応する公開鍵kQを生成する必要がある。

## 【0027】

kの値は次式のように表すことができる。

$$k = (k_0 + k_1 \lambda) \bmod n \quad (3)$$

## 【0028】

上式で、nはE( $F_p$ )上の点の数であり、k<sub>0</sub>およびk<sub>1</sub>は整数である。この場合、点k・Qは次式のようになる。

$$k \cdot Q = (k_0 Q + k_1 \lambda Q) \bmod n \quad (4)$$

## 【0029】

いくつかの暗号演算の場合、kの値をランダムに選択することができ、このような場合、kを選択するのではなく、符号ビットを含まない長さ $\lceil \log_2(n) \rceil / 2$ の、k<sub>0</sub>およびk<sub>1</sub>の値をランダムに選択し（すなわち、k<sub>1</sub>の長さとして、長さkの少なくとも2分の1の長さが選択される）、次いで式(3)を使用してkの値を算出することが可能である。図2に54で示すようにk<sub>0</sub>、k<sub>1</sub>の値を選択した後、Me nezesらによって「Handbook of Applied Cryptography」(HAC)に記載された「Simultaneous Multiple Exponentiation」(アルゴリズム14.88)に類似しており、56に示されているアルゴリズムを使用して式(4)の右辺を高速に計算することができる。都合上、このアルゴリズムを以下に再現する。加群において、べき乗が加算に類似しており、したがって、このアルゴリズム内の乗算を加算で置き換えると以下の手順が得られる。

## 【0030】

アルゴリズム1 同時多重加算

入力：群要素g<sub>0</sub>、g<sub>1</sub>、…g<sub>|t|-1</sub>および負でないtビット整数e<sub>0</sub>、e<sub>1</sub>、…e<sub>|t|-1</sub>。

出力：g<sub>0</sub>e<sub>0</sub> + g<sub>1</sub>e<sub>1</sub> + … + g<sub>|t|-1</sub>e<sub>|t|-1</sub>。

ステップ1。事前計算。 $i$ が0から $(2^l - 1)$ に対して、

【数2】

$$G_i \leftarrow \sum_{j=0}^{l-1} g_j i_j$$

上式で、 $i = (i_{l-1} \dots i_0)_2$ である。

ステップ2。 $A \leftarrow 0$

ステップ3。 $i$ が1から $t$ に達するまで、以下のことを実行する。

$A \leftarrow A + A$ 、 $A \leftarrow A + G_i$

ステップ4。(A)を返す。この場合、 $A = g_0e_0 + g_1e_1 + \dots + g_{l-1}e_{l-1}$

【0031】

このアルゴリズムを数式(4)に適用すると、2つの群要素 $g_0, g_1$ 、すなわち $Q$ および $\psi(Q)$ があり、したがって、2つの整数 $e_0, e_1$ 、すなわち $k_0, k_1$ があることがわかる。このアルゴリズムでは、いくつかの値を事前に算出することができ、最初に $G_i$ が算出される。 $l = 2$ によって $G_i$ を事前に算出した結果をテーブル1(表1)に示す。

【表1】

$i$	0	1	2	3
$G_i$	0	$g_0$	$g_1$	$g_0 + g_1$

【0032】

点加算： $(Q + \psi(Q))$ を実行して点を作成した後、算出済みの要素をテーブル1に記入してテーブル2(表2)を作成することが可能である。図2のステップ58に示すように、これらの要素を事前に算出しメモリに記憶することができる。

【表2】

$i$	0	1	2	3
$G_i$	0	$Q$	$\psi(Q)$	$Q + \psi(Q)$

$G_{1j}$ を求め、したがって、60に示すように $i_1$ から $i_t$ を求めるいかぎり、アルゴリ

ズムのステップを実行することはできない。 $k_i$ の2進表現を使用して概念行列または組合せテーブルを作成することができる。たとえば、 $k_0 = 30$ であり $k_1 = 10$ である場合、 $k_0$ から $k_1$ の2進表現内の最大ビット数が5であるので $t$ は値5を有する。 $k_0$ から $k_1$ の2進表現で作成された概念行列をテーブル3に示す。 $|_i$ は、第1の行が最下位ビットを含み、第2の行が次の下位ビットを含み、以下同様である $i$ 番目の列に表された数によって決定される。したがって、テーブル3(表3)から、 $|_1 = |_2 = (11) = 3$ 、 $|_3 = (01) = 1$ 、 $|_4 = 3$ 、および $|_5 = 0$ であることがわかる。

#### 【表3】

$i$	1	2	3	4	5
$k_0$	1	1	1	1	0
$k_1$	0	1	0	1	0
$ _i$	1	3	1	3	0

#### 【0033】

このアルゴリズムを完了するのに必要なすべての成分を得ることができ、62に示すようにステップ3の反復が行われる。

#### 【0034】

最初は $A \leftarrow 0$ であり、 $i$ は1に設定される。 $|_i = |_1$ であり、これはテーブル3から1に等しい。したがって、 $G|_1$ は $G_1$ であり、これはテーブル2からQである。したがって、 $i = 1$ の場合の反復によるAの値は $0 + Q = Q$ である。

#### 【0035】

$i = 2$ である次の反復の場合、Aの初期値はQであり、したがって、 $A \leftarrow Q + Q = 2Q$ であり、テーブル3から $|_i = |_2 = 3$ である。したがって、 $G|_2$ はテーブル3から $G_3$ に等しく、すなわち、 $Q + \psi(Q)$ である。

#### 【0036】

したがって、 $A + G|_i$ は $2Q + Q + \psi(Q) = 3Q + \psi(Q)$ と算出される。

#### 【0037】

反復は、5回目の反復まで、すなわち、 $k_{0:1} = k$ 、 $\lambda Q$ の値、すなわち $kQ$ が算出されるまで、テーブル4(表4)に記載された*i*の値ごとに継続される。

【表4】

i	A
1	Q
2	$3Q + \psi(Q)$
3	$7Q + 2\psi(Q)$
4	$15Q + 5\psi(Q)$
5	$30Q + 10\psi(Q)$

## 【0038】

各反復では、点倍加 ( $A+A$ ) および点加算 ( $A+G_i$ ) が必要である。ただし、場合によっては、 $G_i$  の値は 0 であり、この場合、計算量が削減される。

## 【0039】

したがって、この方法では、 $\max[\log_2(k_i)]$  に等しい数の点倍加と、ほぼ同数の点加算が必要であることがわかる。点加算の数は、ウィンドウ技法（アルゴリズム14.85 HAC）および指数再符号化技法を使用して削減することができる。 $i$  および  $G_i$  の値を事前に算出することができるので、事前に算出された適切な要素  $G_i$  をテーブル 2 から検索することによって点加算を容易に実行することができる。kP が算出された後、チャネル 16 を介した暗号化伝送または署名伝送において通信者 12 の短命公開鍵としてこの kP を使用することができる。

## 【0040】

簡単に言えば、暗号化や DH 署名などの暗号化演算の場合、整数  $k$  が必要であり、対応する公開鍵  $kQ$  が算出される。それぞれ、長さ  $n$  の 2 分の 1 の長さを有する、値  $k_0$  および  $k_1$  がランダムに選択され、適切なアルゴリズムを使用して項  $k_0Q = k_1\lambda Q$  が生成される。 $k$  をこのように選択すると、この方法は、 $k$  自体をランダムに生成するのと同程度に安全であると思われる。もちろん、効率を向上するためにより少ないビット数の  $k_i$  を選択することが可能である。

## 【0041】

上記の技法において、 $k = k_0 + k_1\lambda$  を書き込み、同時に組み合わせる方法によって、同時多重加算アルゴリズムの速度が高められる。 $k = k_0 + k_1\lambda$  を書き込む技法は、スカラー乗算技法、すなわち、ワインディング、組合せなどと共に使用することもでき有利である。

## 【0042】

いくつかの写像 $\psi$ では、3つ以上の下位要素 $k$  (sub k) を使用することも可能である。いくつかの写像 $\psi$ では、 $k = k_0 + k_1 \lambda + k_2 \lambda^2$ を書き込み、同時多重加算アルゴリズムを適用することによって $k$ の値を算出することができる。

## 【0043】

本発明の第2の実施形態では、一般化された楕円曲線式(1)の異なる形式、すなわち、

$$y^2 = (x^3 - ax) \bmod p \quad (5)$$

が使用される。この場合も、 $p$ は少なくとも160ビットを有する素数である。この種の曲線の場合、 $\gamma$ に必要な特性が異なる。この場合、 $\gamma^2 = -1 \bmod p$ になるような値を求める必要がある。 $\gamma$ の特性を変更するには、異なる写像関数 $\psi'$ を使用する必要がある。この実施形態では、写像は形式 $\psi' : (x, y) \rightarrow (-x, \gamma y)$ をとる。 $(x, y)$ が曲線上にある場合は $\psi'(x, y)$ も曲線上にある。この場合、 $\lambda^4 \equiv 1 \bmod n$  ( $n$ はこの場合も、 $E(F_p)$ 上の点である)、したがって $\lambda$ を算出することができる。写像 $\psi'(Q) = \lambda \cdot Q$ は前述のように実行され、この場合も、この曲線に対して $\lambda$ による乗算を非常に効率的に行うことができる。この実施形態の $k$ に関する数式は、第1の実施形態と同じであり、次式によって表される。

$$k = (k_0 + k_1 \lambda) \bmod n \quad (6)$$

この数式は、前の実施形態と同じであり、2つの群要素のみを有する。したがって、アルゴリズム1の群要素 $Q$ および $Q + \psi'(Q)$ を使用して点 $k \cdot Q$ を算出することができる。この計算では、 $\max\{\log_2(k_i)\}$ に等しい数の点倍加と、同様な数の点加算が必要である。前述のように、ウインドウ技法および指數再符号化技法を使用して点加算の数を削減することができる。

## 【0044】

この方法は、効率的に算出できる自己準同形写像 $\psi$ が存在するかぎり、他の楕円曲線にも適用することができる。

## 【0045】

上記の実施形態は、 $k$ をランダムに選択することができ、したがって、 $k_0$ およ

び $k_1$ を選択することができるものと仮定し、 $k$ を求めている。 $k$ を選択する事が可能な暗号化プロトコルの場合、まず、 $k = (k_0 + k_1 \lambda) \bmod n$ になるように $k$ の所与の値から所望の「短い」形式の $k_0, k_1$ を求める必要がある。場合によつては、3つ以上の $k$ を使用することができ有利である。

#### 【0046】

上述の実施形態でわかるように、事前に点がわかっているときは、テーブルを作成して乗算の速度を高めることができる。しかし、未知の点の倍数が必要になる場合があり（たとえば、これはECDSA検証で起こる可能性がある）、その場合、与えられた $k$ の値をとり、次いで $k_j$ の適切な表現を決定する必要がある。

#### 【0047】

したがって、第3の実施形態では、システム・パラメータおよび値 $k$ が与えられ、点 $Q$ 、必要な倍数 $k$ 、および複素乗算倍数 $\lambda$ が既知である。所定の $k$ の値から「短い」 $k_j$ を求める必要がある。これを行う方法を以下に説明し、図3のフローチャートに示す。（ $k$ を必要としない）事前計算として、 $a_i$ および $b_i$ が $n$ よりも小さな数になるような2つの関係を算出する。

$$a_0 + b_0 \lambda = 0 \bmod n$$

$$a_1 + b_1 \lambda = 0 \bmod n$$

$a_i$ および $b_i$ はできるだけ小さいことが好ましいが、本発明の方法は、 $a_i$ および $b_i$ が最小限でないときでも有利である。 $a_i$ および $b_i$ が共に小さな対 $a_i$ および $b_i$ は、小さなユークリッド長を有するベクトル $u_i$ とみなすことができる。通常、後述の方法では、最初の $k$ のサイズの2分の1の表現を有する $k_0$ および $k_1$ が生成される。

#### 【0048】

本実施形態では、事前に算出された短いベクトル表現を使用して以下の形式の数式を得ることによって、 $kQ$ を効率的に算出することができる。

$$k_0 Q + \lambda k_1 Q$$

#### 【0049】

これは、事前に算出されたベクトルを使用して、 $k$ の知識を必要としない分数 $f_0$ および $f_1$ を導くことによって行われる。ベクトル $z$ は、分数 $f_0$ および $f_1$ と $k$ を

組み合わせることによって生成される。ベクトル $\mathbf{z}$ を使用して、 $\mathbf{v}' = (v_0', v_1')$ である第2のベクトル $\mathbf{v}'$ が算出され、 $kQ$ の値が、

【数3】

$$v_0'Q + \lambda v_1'Q \quad (8)$$

として算出される。この解を得る方法を以下に詳しく説明する。

【0050】

小さな $a_i$ および $b_i$ を作成する場合、 $L_3$ 、すなわち、短い基底ベクトルが直接得られる格子基底削減アルゴリズム（HAC、118ページ）を利用することが可能である。しかし、この好ましい実施形態では、簡単な拡張ユークリッド・アルゴリズムが対 $(n, \lambda)$ に対して使用される。 $(n, \lambda)$ に対する拡張ユークリッド・アルゴリズムにより、 $i$ に応じて $r_i$ の表現（たとえば、ビット長）が小さくなり、 $c_i$ および $d_i$ の表現が大きくなる線形組合せ $c_i n + d_i \lambda = r_i$ が作成される。

【0051】

拡張ユークリッド・アルゴリズムを使用した結果として得られた $| (d_i, r_i) |$ の2つの最小値が保存される。これらのベクトルのサイズが、平方ユークリッド基準 $| (d_i, r_i) | = d_i^2 + r_i^2$ を用いて測定される。これらの最小関係の項は、

$$\hat{d}_0, \hat{r}_0$$

および

$$\hat{d}_1, \hat{r}_1$$

として示され、通常、アルゴリズムの中央で得られる。最小関係が保持されない場合でも、この方法は、部分最適関係により、依然として点倍数の計算において有利である。

【0052】

$a_i$  および  $b_i$  の値は、すべて事前に算出することのできる

$$a_0 = -\hat{r}_0, b_0 = \hat{d}_0$$

および

$$a_1 = -\hat{r}_1, b_1 = \hat{d}_1$$

を定義することによって作成することができる。

#### 【0053】

次のタスクは、倍数  $k$  の小さな表現を求めることである。

#### 【0054】

$a_0, b_0$  および  $a_1, b_1$  の計算が与えられた場合、 $u_0 = (a_0, b_0)$  であり  $u_1 = (a_1, b_1)$  であるベクトル  $u_0, u_1$  を指定することが可能である。これらのベクトルは  $a_i + b_i \lambda = 0 \pmod{n}$  を満たす。群要素  $Q$  にベクトル  $v = (v_0, v_1)$  を乗じることは  $(v_0 + v_1 \lambda)Q$  として定義される。 $a_i + b_i \lambda = 0 \pmod{n}$  であるので、任意の群要素  $R$  について  $u_0R = u_1R = 0$  が成立する。したがって、任意の整数  $z_0$  および  $z_1$  の場合、任意の群要素  $R$  について  $v' R = (v - z_0u_0 - z_1u_1)R$  が成立する。

#### 【0055】

整数  $z_0$  および  $z_1$  としては、ベクトル  $v' = v - z_0u_0 - z_1u_1$  ができるだけ小さな成分を有するような整数が選択される。この場合も、この方法は、 $v'$  の成分が小さい場合に利点を有し、該成分が必ずしも最小限でない場合でも利点を有する。

#### 【0056】

適切な  $z_0$  および  $z_1$  は、 $v$  の基底を正規直交基底  $\{u_0, u_1\}$  に変換することによって算出される。基底間の変換では行列乗算が行われる。ベクトル  $v = (v_0, v_1)$  を  $\{u_0, u_1\}$  基底から標準基底  $\{(1, 0), (0, 1)\}$  に変換する場合、次式が成立する。

#### 【数4】

$$\nu_{\{(1,0),(0,1)\}} = \nu_{(u_0, u_1)} M = (\nu_0, \nu_1) \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$$

他方の方向、すなわち、正規直交基底  $\{(1, 0), (0, 1)\}$  から  $(u_0, u_1)$  基底に変換する場合、乗算は単に  $M$  の逆数によって行われる。

### 【数5】

$$\nu_{(u_0, u_1)} = \nu_{\{(1,0),(0,1)\}} \text{inverse}(M) = \nu_{\{(1,0),(0,1)\}} \frac{1}{a_0 b_1 - a_1 b_0} \begin{bmatrix} b_1 & -b_0 \\ -a_1 & a_0 \end{bmatrix}$$

### 【0057】

ベクトル  $v = (k, 0)$  がゼロ成分を有するので、逆数 ( $M$ ) の下の行は必要とされず、したがって、 $\{u_0, u_1\}$  に変換する場合に必要なのは次式の分数だけである。

### 【数6】

$$f_0 = \frac{b_1}{a_0 b_1 - a_1 b_0}$$

および

### 【数7】

$$f_1 = \frac{b_0}{a_0 b_1 - a_1 b_0}$$

### 【0058】

分数  $f_0$  および  $f_1$  の演算を乗算のみで行うことができるよう、これらの分数を十分な精度に事前に算出しておくことができる。これらの分数をもたらす計算が  $k$  に依存せず、したがって、橍円曲線がシステム・パラメータとして選択される

際にこれらの分数を1度算出することができ、各kごとに再計算する必要はない。同様に、ベクトルv、u<sub>0</sub>、およびu<sub>1</sub>を事前に算出し記憶することができる。

#### 【0059】

kの値が選択されるか、あるいは求められた後、まず、zが(z<sub>0</sub>, z<sub>1</sub>) = (round(kf<sub>0</sub>), round(kf<sub>1</sub>))として定義されるz = (z<sub>0</sub>, z<sub>1</sub>)を計算することによってkQの値を算出することができる。zの近傍の他のベクトルも有用であり、したがって、丸めを床関数または天井関数、あるいは何らかの他の近似で置き換えることができる。

#### 【0060】

適切なzが求められた後、v' = (v<sub>0'</sub>, v<sub>1'</sub>) = v - z<sub>0</sub>u<sub>0</sub> - z<sub>1</sub>u<sub>1</sub>によってv(k, 0)の効率的な等価物が算出される。「効率的な等価物」の句は、v' P = vPおよびv'が小さな係数を有するようなベクトルv'を意味する。この場合、値kQはv<sub>0'</sub> Q + v<sub>1'</sub> λ Qとして算出される。この値は、上述のように同時点加算を使用して算出することができ、上記で説明し、かつH.A.C.14.7の627ページに記載されているように非隣接形式(NAF)再符号化を使用することによってより高い効率を得ることができる。したがって、kが所定の値である場合でも、k<sub>0</sub>およびk<sub>1</sub>の値を算出し写像関数と共に使用してkQの値、したがって、鍵対k、kQを得ることができる。

#### 【0061】

kを3つの部分k = k<sub>0</sub> + k<sub>1</sub>λ + k<sub>2</sub>λ<sup>2</sup>に分離する場合、L<sup>3</sup>行削減によって、次式のような小さなベクトルを得ることができる。

#### 【数8】

$$\begin{bmatrix} 1 & 0 & -\lambda^2 \\ 0 & 1 & -\lambda \\ 0 & 0 & -n \end{bmatrix} \text{ to } \begin{bmatrix} u_2 \\ u_1 \\ u_0 \end{bmatrix}$$

#### 【0062】

二次元の場合と同様に小ベクトル等価物(三次元行)を得ることができる。

## 【0063】

これら の方法を使用して  $k \cdot Q$  の値を求める と、暗号化プロセッサ12によって必要とされる処理能力が大幅に低減される。また、このような反復計算が行われる速度が高められ、それによって、情報を転送するための時間が短縮される。

## 【0064】

当然のことながら、スカラー倍数  $k$  が短縮成分  $k = k_0 + k_1\lambda + k_2\lambda^2 + \dots + k_{m-1}\lambda^{m-1}$  で表された後、同時多重加算アルゴリズムの代わりに、あるいはこのアルゴリズムと共に、効率的な楕円曲線スカラー乗算を行うための他の方法を使用することができる。このような方法には、ウインドウ技法（固定およびスライド）、組合せ技法、ビット再符号化技法、およびこれらの技法の組合せが含まれる。

## 【0065】

特に有益なある技法では、乗算の一成分、たとえば  $k_0$  に関して作成されたテーブルを他の成分  $k_1$  などに再使用することができる。これは、必要に応じて写像  $\gamma$  を適用することにより、算出されたテーブル要素を変換することによって行われる。

## 【0066】

他の例として、 $k$  を  $k = k_0 + k_1\lambda + k_2\lambda^2$  として再設定することができ、 $k$  が  $m$  ビットを有し、 $k_j$  が約  $m/3$  ビットを有する実施形態について以下に説明する。

## 【0067】

成分  $k_j$  は、求められた後、2進表現から、より少ない非ゼロ・ビットを有する符号付き2進表現に再符号化することができる。この再符号化は、表現  $k_j$  内のある 1 ビットまたは -1 ビットが符号付き2進文字列内で他の非ゼロに隣接しないような非隣接形式 (NAF) をとることができる。

## 【0068】

各  $k_j$  が再符号化された後、 $k_j\lambda^{jP}$  の計算を助けるためにテーブルを作成することができる。

## 【0069】

NAFウィンドウ・テーブルは、 $\lambda^i P$ のある短ビット長倍数を事前に算出するためのテーブルである。このウィンドウの幅によってテーブルのサイズが決定される。 $k_1$ が、隣接する非ゼロを有さないように再符号化されているので、奇数ウィンドウ幅が適切である。3ビット幅NAFウィンドウは以下のものを含む。

#### 【数9】

1	1 0 1	1 0 - 1
---	-------	---------

#### 【0070】

再符号化 $k_1$ は、このようなウィンドウを連結し、必要に応じてゼロを充填することによって作成される（H.A.C.、616ページ）。

#### 【0071】

あらゆる非ゼロ・ビットではなく、発生するあらゆるウィンドウごとにのみEC点を加算または減算するだけでよいので、このテーブルを使用することによって必要な加算数を削減することができる。

#### 【0072】

したがって、まず、この技法は $k_0 P$ の計算に適用される。

#### 【0073】

$k_0 P$ の計算のために作成されたテーブルは、演算子 $\gamma$ を使用してテーブル要素に $\gamma$ 写像が写像される場合に $k_1 \lambda P$ 計算に適用することができる。同様に、 $k_0 P$ 用に作成されたテーブルを使用するが、テーブル要素に $\gamma^2$ を写像することによって、 $k_2 \lambda^2 P$ の速度を高めることができる。

#### 【0074】

スライド・ウィンドウ技法を該成分に適用する際に、実行する必要があるのは1組の倍加だけである。

#### 【0075】

好ましい実施形態のこの例を示すために、以下の例を使用する。

#### 【0076】

$k = [101101011101]_2 + [111010101101]_2 \lambda$  である場合、

以下の再符号化を行う。

$$k = [10-100-10-100-101] + [1000-10-10-10-101] \lambda$$

$$= k'_0 + k'_1 \lambda$$

#### 【0077】

1・P、[10-1]・P、[101]・Pを含む、Pに関する3ビット・ウィンドウ・テーブルが事前に算出される。この場合、2回のEC加算および2回のEC倍加が必要である。

#### 【0078】

この後、テーブルの要素を加算／減算することによって、kPを以下のように算出することができる。

$$kP = [10-100-10-100-101]P + [1000-10-10-10-101] \cdot \lambda P$$

これは、累算器Aを使用して以下のように行うことができる。

A←0 : 初期設定

A +=  $\psi(1 \cdot P)$  ;  $k'_0$  の上位ビットを消費する。

A ← 2A ; Aを倍加する。

A ← 2A

A ← [10-1]P ;  $k'_0$  の上位3ビットを消費する。

A ←  $2^4A$

A -= [101] $\psi P$  ;  $k'_1$  の3ビット・ウィンドウを消費する。

A ← 2A ; Aを倍加する。

A -= [101]P ;  $k'_1$  の3ビットを消費する。

A ←  $2^4A$

A -= [101] $\psi P$  ;  $k'_1$  の3ビットを消費する。

A ←  $2^2A$

A -= [10-1]P ;  $k'_0$  の最後のビットを消費する。

A +=  $\psi P$  ; kPを生成する。

#### 【0079】

簡単に言えば、前述の技法は以下のとおりである。橢円曲線Eおよび自己準同

形写像 $\psi$ が与えられた場合、すべての点 $Q \in E$ について $\lambda Q = \psi(Q)$ になるような対応する整数 $\lambda$ が存在する。整数 $m$ を選択し、同等な数 $m$ の「短基底ベクトル」 $b_1, b_2, \dots, b_m$ を算出する。このような基底ベクトルはそれぞれ整数に対応し、そのような整数はそれぞれ、点の数 $n = \#E(F_{p^m})$ （すなわち、点の数）で除することができる。次に、整数 $k$  ( $0 < k < n$ ) が与えられた場合、 $k_i$ として「短い」ベクトルが選択される $k = \sum k_i \cdot \lambda^i$ と書くことができる。これは、 $b_1, b_2, \dots, b_m$ によって生成された格子内の（ $k$ を表す）あるベクトルと近傍のベクトルとの差を求めることによって行われる。

#### 【0080】

以下の実施形態では、複合体の上に画定された橢円曲線に前述の技法（自己準同形写像および基底変換および「Shamir's trick」）を適用することについて明白に説明する。特に、 $p$ が奇素数である曲線 $E(F_{p^m})$ への適用について説明する。以下の実施形態ではこのような曲線に対する技法を例示する。

#### 【0081】

この技法について、写像 $\psi$ がフロベニウス写像 $\psi(x, y) = (x^p, y^p)$ であり、 $A, B \in F_p$ である $E' A, B(F_{p^m})$ が使用される場合において説明する。

#### 【0082】

この場合、フロベニウス写像が $\psi^2 - t\psi + p = 0$ を満たし、 $t = p + 1 - \#E(F_{p^m})$ であることがわかっている。

#### 【0083】

したがって、 $\lambda^2 - t\lambda + p = 0 \pmod{n}$ であり、また $\lambda^{2+i} - t\lambda^{1+i} + p\lambda^i = 0 \pmod{n}$ である。

#### 【0084】

以下のベクトルが、ベクトル空間 $Q_n$ 空間の $m$ 個の「短」基底ベクトルで構成されている。

#### 【数10】

$$\begin{array}{ll}
 & (\lambda^{m-1}, \dots, \lambda^2, \lambda^1, \lambda^0) \\
 b_1 & (0, 0, 0, \dots, 0, 1, -t, p) \\
 b_2 & ( \quad \quad \quad \quad \quad 1, -t, p, 0) \\
 \\ 
 & (1, -t, p, 0, 0, \dots, 0) \\
 & (-t, p, 0, 0, \dots, 0, 1) \\
 b_m & (p, 0, 0, 0, \dots, 0, 1, -t)
 \end{array}$$

したがって、このような曲線に対する  $k \cdot Q$  を計算する場合、ベクトル  $b_1, b_2, \dots, b_m$  および前述の技法を使用することができる。

#### 【〇〇八五】

上記の実施形態では、 $k, \lambda Q$  を  $\psi(kQ)$  から得ることができ、写像が加算よりも効率的であることが理解されよう。

## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

Int'l. Appl. No.  
PCT/CA 99/01222

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC 7 G06F7/72		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC 7 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KOBLITZ N: "CM-CURVES WITH GOOD CRYPTOGRAPHIC PROPERTIES" PROCEEDINGS OF THE CONFERENCE ON THEORY AND APPLICATIONS OF CRYPTOGRAPHIC TECHNIQUES (CRYPTO), DE, BERLIN, SPRINGER, vol. -, 1991, pages 279-287, XP000269035 page 280	1-8
A	V. MÜLLER: "Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two" -, 'Online' 30 June 1997 (1997-06-30), pages 1-19, XP002121579 Retrieved from the Internet: <URL:ftp://ftp.informatik.tu-darmstadt.de/pub/TI/reports/vmueller.jc.ps.gz> 'retrieved on 1999-10! the whole document	1-8
	-/-	
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.		<input type="checkbox"/> Patent family members are listed in annex.
<p>* Special categories of cited documents:</p> <ul style="list-style-type: none"> <li>*A* document defining the general state of the art which is not considered to be of particular relevance</li> <li>*E* earlier document but published on or after the International filing date</li> <li>*U* document which may throw doubt on priority, claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</li> <li>*C* document referring to an oral disclosure, use, exhibition or other means</li> <li>*P* document published prior to the International filing date but later than the priority date claimed</li> </ul>		
Date of the actual completion of the international search		Date of mailing of the international search report
2 May 2000		15/05/2000
Name and mailing address of the ISA European Patent Office, P.O. Box 3020, Postbus 2 NL - 2280 HV Rijswijk Tel: (+31-70) 340-2040, Fax: 31 651 890 16 Fax: (+31-70) 340-3016		Authorized officer Verhoof, P

1

## INTERNATIONAL SEARCH REPORT

Inte  
onal Application No  
PCT/CA 99/01222

C(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Character of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CHEON J H ET AL: "Two efficient algorithms for arithmetic of elliptic curves using Frobenius map" PUBLIC KEY CRYPTOGRAPHY. FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98. PROCEEDINGS, PUBLIC KEY CRYPTOGRAPHY FIRST INTERNATIONAL WORKSHOP ON PRACTICE AND THEORY IN PUBLIC KEY CRYPTOGRAPHY, PKC'98 PROCEEDINGS, , pages 195-202, XP000905121 1998, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-64693-0 the whole document	1-8
A	SOLINAS J A: "An improved algorithm for arithmetic on a family of elliptic curves" ADVANCES IN CRYPTOLOGY - CRYPTO '97. 17TH ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE. PROCEEDINGS, ADVANCES IN CRYPTOLOGY - CRYPTO'97. 17TH ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE. PROCEEDINGS, SANTA BARBARA, CA, USA, 17-21 AUG. 1997, pages 357-371, XP002136758 1997, Berlin, Germany, Springer-Verlag, Germany ISBN: 3-540-63384-7 the whole document	1-8

1

Form PCT/GB/0210 (continuation of second sheet) (July 1992)

page 2 of 2

---

**フロントページの続き**

(81)指定国 EP(AT, BE, CH, CY,  
DE, DK, ES, FI, FR, GB, GR, IE, I  
T, LU, MC, NL, PT, SE), OA(BF, BJ  
, CF, CG, CI, CM, GA, GN, GW, ML,  
MR, NE, SN, TD, TG), AP(GH, GM, K  
E, LS, MW, SD, SL, SZ, TZ, UG, ZW  
, EA(AM, AZ, BY, KG, KZ, MD, RU,  
TJ, TM), AE, AL, AM, AT, AU, AZ,  
BA, BB, BG, BR, BY, CA, CH, CN, C  
U, CZ, DE, DK, EE, ES, FI, GB, GD  
, GE, GH, GM, HR, HU, ID, IL, IN,  
IS, JP, KE, KG, KP, KR, KZ, LC, L  
K, LR, LS, LT, LU, LV, MD, MG, MK  
, MN, MW, MX, NO, NZ, PL, PT, RO,  
RU, SD, SE, SG, SI, SK, SL, TJ, T  
M, TR, TT, UA, UG, US, UZ, VN, YU  
, ZA, ZW

(72)発明者 ヴァンストーン、スコット、エー.  
カナダ国 エルオピー 1ピー0 オンタ  
リオ州、キャンプベルヴィル、ピー. オ  
ー. ボックス 490、パインビュー ト  
レイル 10140

Fターム(参考) 5J104 AA18 AA25 JA25 NA02 NA16  
NA35 NA40

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USP70)**